**Phase 3: Development part 1**

**Collect Traffic Data:**

Write Python scripts to collect traffic data using the appropriate sensors or cameras. Depending on the type of sensors, this could involve interfacing with GPIO pins, USB cameras, or other hardware.To deploy IoT devices and develop a Python script for sending real-time traffic data to a traffic information platform, follow these steps:

1. Choose IoT Devices:

Select appropriate IoT devices such as traffic flow sensors, cameras, or any other sensors needed for your application.

2. Hardware Setup:

Install and configure the IoT devices at strategic locations, ensuring they have power sources and internet connectivity.

3. Set Up a Traffic Information Platform:

Establish a traffic information platform or use an existing one to receive and process the data. This could be a cloud-based server or a dedicated application.

4. Python Script Development:

Develop a Python script to collect and send real-time traffic data. You'll need to customize this to fit your specific devices and platform:

**Send Data to the Traffic Information Platform:**

You need a way to send the collected data to the Traffic Information Platform. You can use various methods such as HTTP requests, MQTT, or other IoT protocols. Here's an example of how to send data using HTTP:

import requests

# Replace with your Traffic Information Platform's API endpoint

api\_url = "https://your-traffic-api.com/data"

# Data to send (modify this according to your data format)

traffic\_data = {

"location": "Intersection A",

"vehicle\_count": 100,

"average\_speed": 45,

# Add more data fields as needed

}

# Send data using a POST request

response = requests.post(api\_url, json=traffic\_data)

if response.status\_code == 200:

print("Data sent successfully")

else:

print("Failed to send data. Status code:", response.status\_code)

**Handle Data on the Traffic Information Platform:**

Set up the Traffic Information Platform to receive and process incoming data. You will need a web server, database, and data processing logic on this platform to store and analyze the data. Python, along with web frameworks like Flask or Django, can be used for building the server-side logic.

**Team Members:**

A.Nandhini(620821205037)

T.Kanimozhi(620821205024)

M.Kaviya(620821205029)

S.Deepika(620821205012)